

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-31147

(43)公開日 平成11年(1999) 2月2日

(51)IntCl.⁶

G 0 6 F 17/30

識別記号

F I

G 0 6 F 15/419

3 1 0

審査請求 未請求 請求項の数3 O L (全 8 頁)

(21)出願番号 特願平9-185407

(22)出願日 平成9年(1997) 7月10日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 伊藤 有紀

東京都府中市東芝町1番地 株式会社東芝
府中工場内

(74)代理人 弁理士 三好 秀和 (外3名)

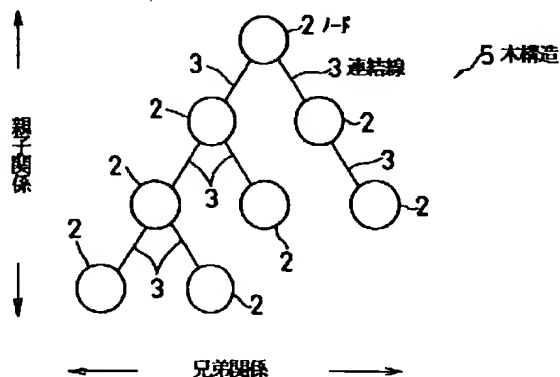
(54)【発明の名称】 木構造を有するデータ管理システム

(57)【要約】

【課題】 既に構成されている木構造に対し、新たなノードを挿入したり、既に構成されている木構造から不要なノードを削除するとき、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了し得るようにする。

【解決手段】 木構造5を構成する各ノード2に対し、全ての番号が一意になるように、ノード番号を付加するとともに、各ノード番号の関係を、各ノード2毎にまとめて、関係記述子4を作成し、これらの関係記述子4をまとめて、関係記述テーブル10を作成し、各ノード2の追加や削除を行なう際、関係記述テーブル10の内容を変更して、各ノード2の接続関係を変更する。

1 木構造を有するデータ管理システム



【特許請求の範囲】

【請求項1】 複数のノードを木構造形式で接続して使用する木構造を有するデータ管理システムにおいて、複数のノードを階層化して接続した木構造における自分の親、自分の第1子、自分と同じ親を持つメンバーとなる各ノードのノード番号を保持する関係記述テーブルを備え、

各ノードの追加、削除を行なうとき、関係記述テーブルの内容のうち、追加、削除対象となったノードに関連するノードの番号を変更することを特徴とする木構造を有するデータ管理システム。

【請求項2】 複数のノードを木構造形式で接続して使用する木構造を有するデータ管理システムにおいて、複数のノードを階層化して接続した第1木構造と、自分の親、自分の第1子、自分と同じ親を持つメンバーとなる各ノードのノード番号を保持する関係記述テーブルとを備え、

前記第1木構造中にあるノードの1つに、複数のノードを階層化して接続した第2木構造を追加する際には、第2木構造の各ノード番号全体を変更するとともに、追加対象となった前記ノード自体、および関連するノードに基づき、第1木構造に対応する関係記述テーブル、および前記第2木構造に対応する関係記述テーブルの内容を変更することを特徴とする木構造を有するデータ管理システム。

【請求項3】 請求項2に記載の木構造を有するデータ管理システムにおいて、

前記第1木構造中にあるノードの1つから下にある各ノードを削除する際、第1木構造の関係記述テーブル上にある各関係記述子のうち、削除した各ノードに対応する各関係記述子を削除するとともに、削除した各ノードのノード番号を持つ関係記述子の内容を変更することを特徴とする木構造を有するデータ管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、管理構成要素などのデータ要素を木構造にして所有する木構造を有するデータ管理システムに関する。

【0002】

【従来の技術】管理構成要素などのデータ要素を木構造にして所有する木構造を有するデータ管理システムでは、従来、図15に示すように、複数のノード101によって構成される木構造102に対し、図16に示すように、一番上のノード101から順次、番号を付けて、各ノード101に、上下関係（親子関係）を与えるとともに、各データ要素の上下関係に応じて、各データ要素を各ノード101に割り付けることにより、1つのデータ要素が与えられたとき、このデータ要素の親となるデータ要素や子となるデータ要素などを順次、見つけ出すことができるようにしている。

【0003】

【発明が解決しようとする課題】しかしながら、上述した従来の木構造を有するデータ管理システムにおいては、既に構成されている木構造102に新たなデータ要素を挿入したときや既に構成されている木構造102から不要になったデータ要素を削除したとき、一番上のノード101から、順次、番号を付け直して、全てのノード101に与えられた番号を付け直さなければならないことから、木構造102の再構成が難しいという問題があった。

【0004】本発明は上記の事情に鑑み、請求項1では、既に構成されている木構造に対し、新たなノードを挿入したり、既に構成されている木構造から不要なノードを削除するとき、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に短くすることができる木構造を有するデータ管理システムを提供することを目的としている。

【0005】また、請求項2では、既に構成されている木構造に対し、新たなノードを挿入するとき、新たなノードと、このノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させることができる木構造を有するデータ管理システムを提供することを目的としている。

【0006】また、請求項3では、既に構成されている木構造から1つのノードの下にある各ノードを削除するとき、削除されたノード自体と、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させることができる木構造を有するデータ管理システムを提供することを目的としている。

【0007】

【課題を解決するための手段】上記の目的を達成するために本発明は、請求項1では、複数のノードを木構造形式で接続して使用する木構造を有するデータ管理システムにおいて、複数のノードを階層化して接続した木構造における自分の親、自分の第1子、自分と同じ親を持つメンバーとなる各ノードのノード番号を保持する関係記述テーブルを備え、各ノードの追加、削除を行なうとき、関係記述テーブルの内容のうち、追加、削除対象となったノードに関連するノードの番号を変更することを特徴としている。

【0008】また、請求項2では、複数のノードを木構造形式で接続して使用する木構造を有するデータ管理システムにおいて、複数のノードを階層化して接続した第

1木構造と、自分の親、自分の第1子、自分と同じ親を持つメンバーとなる各ノードのノード番号を保持する関係記述テーブルとを備え、前記第1木構造中にあるノードの1つに、複数のノードを階層化して接続した第2木構造を追加する際には、第2木構造の各ノード番号全体を変更するとともに、追加対象となった前記ノード自体、および関連するノードに基づき、第1木構造に対応する関係記述テーブル、および前記第2木構造に対応する関係記述テーブルの内容を変更することを特徴としている。

【0009】さらに、請求項3では、請求項2に記載の木構造を有するデータ管理システムにおいて、前記第1木構造中にあるノードの1つから下にある各ノードを削除する際、第1木構造の関係記述テーブル上にある各関係記述子のうち、削除した各ノードに対応する各関係記述子を削除するとともに、削除した各ノードのノード番号を持つ関係記述子の内容を変更することを特徴としている。

【0010】上記の構成において、請求項1では、複数のノードを階層化して接続した木構造と、自分の親、自分の第1子、自分と同じ親を持つメンバーとなる各ノードのノード番号を保持する関係記述テーブルとを備え、各ノードの追加、削除を行なうとき、関係記述テーブルの内容のうち、追加、削除対象となったノードに関連するノードの番号を変更することにより、既に構成されている木構造に対し、新たなノードを挿入したり、既に構成されている木構造から不要なノードを削除するとき、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了させ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に短くする。

【0011】また、請求項2では、第1木構造中にあるノードの1つに、第2木構造を追加する際、第2木構造の各ノード番号全体を変更した後、追加対象となった前記ノード自体および関連するノードに基づき、第1木構造に対応する関係記述テーブル、前記第2木構造に対応する関係記述テーブルの内容を変更することにより、既に構成されている木構造に対し、新たなノードを挿入するとき、新たなノードと、このノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了させ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させる。

【0012】さらに、請求項3では、第1木構造中にあるノードの1つから下にある各ノードを削除する際、第1木構造の関係記述テーブル上にある各関係記述子のうち、削除した各ノードに対応する各関係記述子を削除するとともに、削除した各ノードのノード番号を持つ関係記述子の内容を変更することにより、既に構成されている木構造から1つのノードの下にある各ノードを削除す

るとき、削除されたノード自体と、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了させ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させる。

【0013】

【発明の実施の形態】

《実施の形態の構成》図1は本発明による木構造を有するデータ管理システムの実施の形態を示す構成図である。

【0014】この図に示す木構造を有するデータ管理システム1は、親子関係、兄弟関係などに応じた位置に配置される複数のノード2と、各ノード2間を接続する複数の連結線3と、図2に示すように、各ノード2毎に作成される複数の関係記述子4とを備えており、各ノード2に対し、全ての番号が一意になるように、ノード番号が付加されるとともに、各ノード番号の関係が、各ノード2毎にまとめられて、関係記述子4が作成され、これらの関係記述子4がまとめられて、関係記述テーブル10（図3参照）が作成される。そして、新たなノード2を追加する際や不要になったノード2を削除する際に、関係記述テーブル10中にある各ノード番号のうち、追加されたノード2、削除されたノード2に対応するノード番号を付け替えるだけで、木構造5の変更を完了する。

【0015】この場合、複数のノード2と、これらを接続する複数の連結線3とによって、1つの木構造5を構成し、各ノード2に対し、各データ要素が割り付けられる。

【0016】また、関係記述子4は、図2に示すように、自分のノード番号が書き込まれる自分記入欄6と、自分の親となるノード2のノード番号が書き込まれる親記入欄7と、自分の第1子となるノード2のノード番号が書き込まれる第1子記入欄8と、自分と同じ親を持つメンバーのなかで、最も関係が近いノード2のノード番号が書き込まれるメンバー記入欄9とを備えている。また、図3に示すように、各ノード2毎の関係記述子4がまとめられて、1つの関係記述テーブル10が作成される。

【0017】《実施の形態の動作》次に、図4～図14に示す模式図を参照しながら、このデータ管理システム1を構成する木構造5、関係記述テーブル10の作成動作、新たなノード2の挿入動作、不要なノード2の削除動作について詳細に説明する。

【0018】＜木構造、関係記述テーブルの作成動作＞まず、各ノード2にノード番号を付けるときには、図4に示すように、第1木構造5aを構成する各ノード2に対し、ノード番号が一意になるように、ノード番号を付ける。このとき、各ノード番号は必ずしも一番上の親から順番になっていなくても良い。

10

20

30

40

50

5

【0019】この後、図5に示すように、第1関係記述テーブル10a上にある各関係記述子4の中から、自分コード記入欄6に自分のコード番号が書き込まれた関係記述子4を選択するとともに、自分の親となるノード2のノード番号を求め、これを自分の関係記述子4中にある親記入欄7に書き込む。但し、一番上のノード2、例えばノード番号“5”については、自分の親のノード番号として、“0”を書き込む。

【0020】次いで、各ノード2毎に、自分の第1子となるノード2のノード番号を求め、これを自分の関係記述子4中にある第1子記入欄8に書き込む。但し、自分の第1子が無い、一番下のノード2、例えばノード番号“8”、“3”、“9”、“12”、“10”、“11”については、自分の第1子のノード番号として、“0”を書き込む。

【0021】この後、各ノード2毎に、自分と同じ親を持つノード2、すなわちメンバーとなるノード2を求め、このノード2のノード番号を、自分の関係記述子4中にあるメンバー記入欄9に書き込む。但し、自分と同じ親を持つメンバーが複数存在し、かつ各メンバーの中での上下関係において、自分が一番下であるときには、自分と同じ親を持つメンバーのノード番号として、一番上のメンバーのノード番号を書き込む。また、自分の他に、自分と同じ親を持つメンバーがいないノード、例えばノード番号“5”、“7”、“11”については、自分のメンバーとして、“0”を書き込む。

【0022】<新たなノードの挿入動作>次いで、このような手順で、複数のノード2と、複数の連結線3とによって第1木構造5aを作成した後、この第1木構造5aを構成する各ノード2の1つ、例えばノード番号“6”のノード2に対し、新たな木構造、例えば図6に示すような複数のノード2と、複数の連結線3とを持つ木構造（第2木構造5b）を挿入する場合には、上述した第1木構造5aの作成動作と同様な手順で、図7に示すように、この第2木構造5bを構成する各ノード2に対し、ノード番号を与えると同時に、各ノード2の関係に基づき、図8に示す第2関係記述テーブル10bを作成する。

【0023】この後、図9に示すように、第1木構造5aを構成するノード番号“6”のノード2の子として、第2木構造5bの一番上にあるノード2を接続して、第3木構造5cを作成した後、この第3木構造5cを構成する第2木構造5bの部分について、「第2木構造5bを構成する各ノード2の新たなノード番号=第2木構造5bにおける自分の元のノード番号+第1木構造5aの総ノード数」によって示される演算を行なって、第2木構造5bを構成している各ノード2の新たなノード番号を求め、図10に示すように、第2木構造5bを構成している各ノード2に新たなノード番号を与える。

【0024】次いで、図11に示すように、第2木構造

6

5bに対応する元の第2関係記述テーブル10b内にある元のノード番号を新たなノード番号に書き替える。但し、自分の親となるノード2のノード番号、自分の第1子となるノード2のノード番号、自分と同じ親を持つ各メンバーのノード番号が“0”のときには、これを“0”のままに保持する。この後、図12に示すように、書き替え済みの第2関係記述テーブル10bを第1木構造5aに対応する第1関係記述テーブル10aに追加して、第3関係記述テーブル10cを作成する。

【0025】そして、第2木構造5bが追加されたノード2、すなわちノード番号“6”のノード2自体およびこのノード番号“6”に關係するノード番号“11”、“13”を持つ各ノード2について、次に述べるように、第3関係記述テーブル10cの内容を変更する。

【0026】まず、ノード番号“6”の関係記述子4内の第1子記入欄8に記入されているノード番号“11”をノード番号“13”に変更するとともに、ノード番号“11”の関係記述子4内のメンバー記入欄9に記入されている“0”をノード番号“13”に変更する。さらに、ノード番号“13”の関係記述子4内の親記入欄7に記入されている“0”をノード番号“6”に変更するとともに、関係記述子4内のメンバー記入欄9に記入されている“0”をノード番号“11”に変更する。

【0027】これによって、第1木構造5aを構成する1つのノード2に対し、第2木構造5bを追加しても、第1木構造5aを構成する各ノード2のノード番号のうち、第2木構造5bが追加されたノード2、すなわちノード番号“6”のノード2自体およびこのノード番号“6”に關係するノード番号“11”、“13”を持つ各ノード2について、第1関係記述テーブル10a、第2関係記述テーブル10bの内容を変更するだけで、新たなノード2の挿入に伴うノード番号の変更処理を完了させることができる。

【0028】<不要なノード2の削除動作>また、上述した手順で、複数のノード2と、複数の連結線3とによって第1木構造5aを作成した後、この第1木構造5aを構成する各ノード2の1つ、例えばノード番号“4”のノード2から下にある各ノード2を削除する場合には、まず図13に示すように、第1木構造5aを構成する各ノード2のうち、ノード番号“4”から下のノード2を全て削除して第4木構造5dを作成する。

【0029】この後、第1木構造5aに対応する第1関係記述テーブル10a内にある各関係記述子4のうち、削除した各ノード2のノード番号、すなわちノード番号“4”、“7”、“9”、“10”、“12”に対応する関係記述子4を全て削除して、図14に示すような第4関係記述テーブル10dを作成するとともに、この第4関係記述テーブル10d内にある各ノード番号のうち、削除したノード2のノード番号を持つ関係記述子4、例えばノード番号“2”を持つノード2の関係記述

子4を構成するメンバー記入欄9に記入されているノード番号“4”を“0”に変更する。

【0030】これによって、第1木構造5aを構成する1つのノード2から下の各ノード2を削除しても、第1木構造5aを構成する各ノード2のノード番号のうち、削除した各ノード2、すなわちノード番号“4”、“7”、“9”、“10”、“12”を持つ各ノード2について、第1関係記述テーブル10aの内容を変更するだけで、不要なノード2の削除に伴うノード番号の変更処理を完了させることができる。

【0031】《実施の形態の効果》このようにこの実施の形態においては、各ノード2に対し、全ての番号が一意になるようにノード番号を付加するとともに、各ノード番号の関係を各ノード2毎にまとめて関係記述子4を作成し、これらの関係記述子4をまとめて関係記述テーブル10を作成するようにしたので、新たなノード2を追加する際や不要になったノード2を削除する際に、関係記述テーブル10中にある各ノード番号のうち、追加されたノード2、削除されたノード2に対応するノード番号を付け替えるだけで、木構造5の変更を完了することができ、これによって木構造5の再構成作業を簡単にし、オペレータの手間を飛躍的に軽減させることができるとともに、コンピュータ装置側の処理時間を飛躍的に短くすることができる。

【0032】また、この実施の形態では、既に構成されている木構造5に対し、新たなノード2を挿入するとき、新たなノード2と、このノード2に関連するノード2のノード番号を付け替えるだけで、木構造5の変更を完了することができ、これによって木構造5の再構成作業を簡単にし、オペレータの手間を飛躍的に軽減させることができるとともに、コンピュータ装置側の処理時間を飛躍的に短くすることができる。

【0033】同様に、既に構成されている木構造5から1つのノード2の下にある各ノード2を削除するとき、削除されたノード2自体と、各ノード2に関連するノード2のノード番号を付け替えるだけで、木構造5の変更を完了することができ、これによって木構造5の再構成作業を簡単にし、オペレータの手間を飛躍的に軽減させることができるとともに、コンピュータ装置側の処理時間を飛躍的に短くすることができる。

【0034】

【発明の効果】以上説明したように本発明によれば、請求項1では、既に構成されている木構造に対し、新たなノードを挿入したり、既に構成されている木構造から不要なノードを削除するとき、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に短くすることができる。

【0035】また、請求項2では、既に構成されている

木構造に対し、新たなノードを挿入するとき、新たなノードと、このノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させることができる。

【0036】また、請求項3では、既に構成されている木構造から1つのノードの下にある各ノードを削除するとき、削除されたノード自体と、各ノードに関連するノードの番号を付け替えるだけで、木構造の変更を完了することができ、これによって木構造の再構成作業を簡単にし、オペレータの操作時間やコンピュータ装置側の処理時間などを飛躍的に軽減させることができる。

【図面の簡単な説明】

【図1】本発明による木構造を有するデータ管理システムの実施の形態を示す構成図である。

【図2】図1に示す木構造を有するデータ管理システムを構成する関係記述子の一例を示す模式図である。

【図3】図2に示す関係記述子をまとめた関係記述テーブルの一例を示す模式図である。

【図4】図1に示す木構造を有するデータ管理システムの木構造作成動作を説明するための模式図である。

【図5】図1に示す木構造を有するデータ管理システムの関係記述テーブル作成動作を説明するための模式図である。

【図6】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図7】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図8】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図9】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図10】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図11】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図12】図1に示す木構造を有するデータ管理システムのノード挿入動作を説明するための模式図である。

【図13】図1に示す木構造を有するデータ管理システムのノード削除動作を説明するための模式図である。

【図14】図1に示す木構造を有するデータ管理システムのノード削除動作を説明するための模式図である。

【図15】従来から知られている一般的な木構造を有するデータ管理システムの一例を示す構成図である。

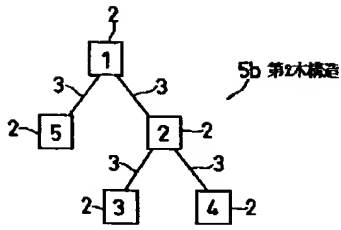
【図16】図1に示す各ノードにノード番号を付加したときの一例を示す模式図である。

【符号の説明】

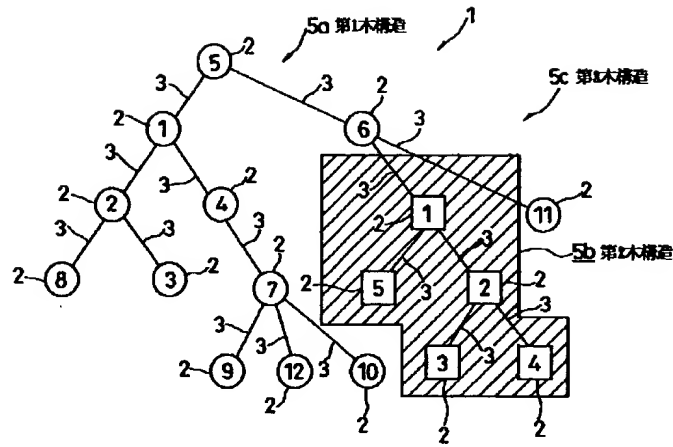
1 データ管理システム

2 ノード

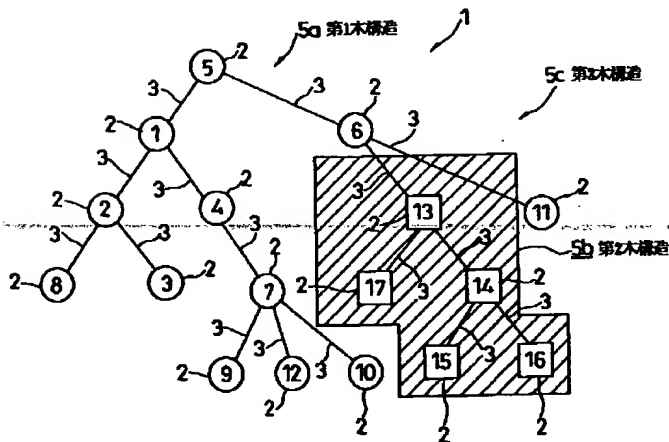
【図7】



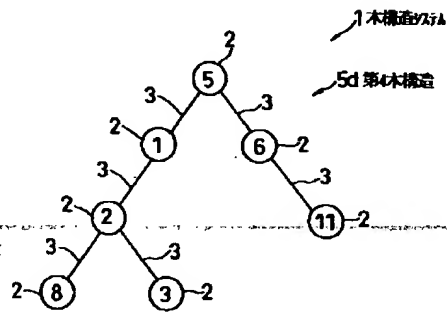
【図9】



【図10】



【図13】



【図11】

| | 6 | 7 | 8 | 9 |
|-----|----|----|----|----|
| 4 { | 13 | 0 | 17 | 0 |
| | 14 | 13 | 15 | 17 |
| 4 { | 15 | 14 | 0 | 16 |
| | 16 | 14 | 0 | 15 |
| 4 { | 17 | 13 | 0 | 14 |

10b 第4関係記述-1/1

【図14】

| | 6 | 7 | 8 | 6 |
|-----|----|---|----|---|
| 4 { | 1 | 5 | 2 | 6 |
| | 2 | 1 | 8 | 5 |
| 4 { | 3 | 2 | 0 | 8 |
| | 5 | 0 | 1 | 0 |
| 4 { | 6 | 5 | 11 | 1 |
| | 8 | 2 | 0 | 3 |
| 4 { | 11 | 6 | 0 | 0 |

10d 第4関係記述-1/1

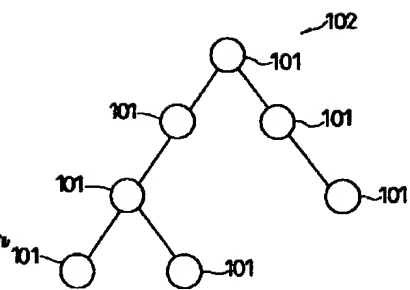
【図12】

1 木構造の例

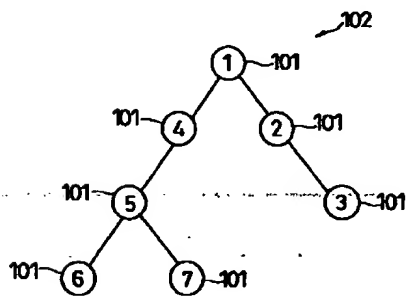
| | 6 | 7 | 8 | 9 |
|-----|----|----|----|----|
| 4 { | 1 | 5 | 2 | 6 |
| | 2 | 1 | 8 | 4 |
| 4 { | 3 | 2 | 0 | 8 |
| | 4 | 1 | 7 | 2 |
| 4 { | 5 | 0 | 1 | 0 |
| | 6 | 5 | 13 | 1 |
| 4 { | 7 | 4 | 9 | 0 |
| | 8 | 2 | 0 | 3 |
| 4 { | 9 | 7 | 0 | 12 |
| | 10 | 7 | 0 | 9 |
| 4 { | 11 | 6 | 0 | 13 |
| | 12 | 7 | 0 | 10 |
| 4 { | 13 | 5 | 17 | 11 |
| | 14 | 13 | 15 | 17 |
| 4 { | 15 | 14 | 0 | 16 |
| | 16 | 14 | 0 | 15 |
| 4 { | 17 | 12 | 0 | 14 |

10a 第1関係記述
10b 第2関係記述
10c 第3関係記述

【図15】



【図16】



PTO: 2003-358

Japanese Published Unexamined Patent Application (A) No. 11-031147, published February 2, 1999; Application Filing No. 9-185407, filed July 10, 1997; Inventor(s): Arinori Ito; Assignee: Toshiba Corporation ; Japanese Title: Data Control System Having A Tree Structure

DATA CONTROL SYSTEM HAVING A TREE STRUCTURE

CLAIM(S)

1) A data control system having a tree structure with multiple nodes connected and a relationship-indicating table containing the node numbers of the nodes which will become, a parent, child, and members sharing the same parent in the tree structure constructed by hierarchically connecting multiple nodes, characterized in that, when each node is added or deleted, the node number of the node to be deleted or added is changed in the content of the relationship-indicating table.

1) A data control system having a tree structure with multiple nodes connected, comprising a first tree structure wherein multiple nodes are hierarchically connected and a relationship-indicating table containing the node numbers of the nodes which will become, a parent, child, and members sharing the same parent in the tree structure constructed by hierarchically connecting multiple nodes, characterized in that, when a second tree structure, wherein multiple nodes are hierarchically connected, is added to one of the nodes in the first tree structure,

all the node numbers in the second tree structure are changed, while simultaneously changing the contents of the relationship-indicating table corresponding to said first tree structure and to said second tree structure based on said node to be added and its related node.

3) A data control system having a tree structure mentioned in Claim 2, characterized in that, when each node under one of the nodes in said first tree structure is deleted, each relationship-indicator corresponding to each deleted node is deleted out of the nodes in the relationship-indicating table in the first tree structure, while changing the content of relationship-indicator having the node number of each deleted node.

DETAILED DESCRIPTION OF THE INVENTION

(0001)

(Field of Industrial Application)

The present invention pertains to a data control system having a tree structure, wherein the data elements such as the control structure elements are formed into a tree structure.

(0002)

(Prior Art)

With the prior art data control system having a tree structure, wherein the data elements such as control structure elements are arranged in form of a tree, to

the tree structure 102 constructed by multiple nodes 10 shown in Fig. 15, numbers are assigned starting from the topmost node 101 to provide a hierarchical relationship to each node (parent and child relationship), as shown in Fig. 16; at the same time, each data element is assigned to each node 101 according to the hierarchical relationship of each data; by so doing, when one data element is provided, a parent data to said data or a child data to said data can be found.

(0003)

(Problems of the Prior Art to Be Solved)

With the prior art data control system having a tree structure, when a new data element is inserted into the existing tree structure 102 or an unnecessary data element is deleted from it, the numbers must be reassigned to all the nodes starting from the topmost node 101, which is difficult in restructuring the tree structure 102.

(0004)

To solve the aforementioned problems, Claim 1 of the present invention claims that, when a new node is inserted into or deleted from the existing tree structure, the change of the tree structure can be completed only by reassigning the number of the node related to each node, which results in simplifying the restructuring operation of the tree structure; thereby presenting a data control system having a tree structure that can dramatically reduce the operation time of the operator and processing time of the computer device.

(0005)

Claim 2 of the present invention claims that, when a new node is inserted into the existing tree structure, the change of the tree structure can be completed only by reassigning the number of the node related to the new node, which results in simplifying the restructuring operation of the tree structure; thereby presenting a data control system having a tree structure that can dramatically reduce the operation time of the operator and processing time of the computer device.

(0006)

Claim 3 of the present invention claims that, when each node under one node is deleted from the existing tree structure, the change of the tree structure can be completed only by reassigning the numbers of the deleted node and of the node related to each node, which results in simplifying the restructuring operation of the tree structure; thereby presenting a data control system having a tree structure that can dramatically reduce the operation time of the operator and the processing time of the computer device.

(0007)

(Means to Solve the Problems)

To solve the aforementioned problems, Claim 1 of the present invention presents a data control system having a tree structure, wherein multiple nodes are connected in form of a tree structure, characterized in that said system has a

relationship-indicating table containing the numbers of the node of a parent to itself, its own first child, and of every node which is a member sharing the same parent with itself, and in that when each node is added or deleted, the number of the node related to the node to be added or deleted in said relationship-indicating table is changed.

(0008)

Claim 2 of the present invention presents a data control system having a tree structure, wherein multiple nodes are connected in form of a tree structure, characterized in that said system has a first tree structure with multiple nodes hierarchically connected and a relationship-indicating table containing the numbers of the nodes of a parent to itself, its own first child, and of each node which is a member sharing the same parent, and in that, when a second tree structure with multiple nodes hierarchically connected is added to one of the nodes in said first tree structure, the number of each node in the second tree structure is changed and, based on said node to be added and its related node, the content of the relationship-indicating table for the first tree structure and that for the second tree structure are changed.

(0009)

Claim 3 of the present invention presents a data control system having a tree structure of Claim 2, characterized in that, when each node under one node in said

first tree structure is deleted, each relationship-indicator corresponding to each deleted node is deleted from each relationship-indicator in the relationship-indicating table of the first tree structure, and the content of the relationship-indicator having the node number of the deleted node is changed.

(0010)

Claim 1 claims that the aforementioned structure comprises a tree structure with multiple nodes hierarchically connected and a relationship-indicating table containing the node numbers of the parent to itself, its own first child, and of members sharing the same parent with itself, and that the change of the tree structure can be completed by merely changing the number of the node related to each node to be added or deleted when a new node is inserted into the existing tree structure or an unnecessary node is deleted from it; thereby simplifying the restructuring operation of the tree structure, and dramatically reducing the operation time of the operator and processing time of the computer device.

(0011)

Claim 2 claims that, when the second tree structure is added to one of the nodes in the first tree structure, after all the node numbers in the second tree structure have been changed, the contents of the relationship-indicating table corresponding to the first tree structure and that of the second tree structure are changed based on the node related to said node to be added. Therefore, when a new

node is inserted into the existing tree, changing of the tree structure can be completed by merely changing the node numbers of the new node and of its related node; thereby simplifying the restructuring operation of the tree structure, and dramatically reducing the operation time of the operator and the processing time of the computer device.

(0012)

Claim 3 claims that, when each node under one node in the first tree structure has been deleted, each relationship-indicator corresponding to each deleted node is deleted from the relationship-indicators in the relationship-indicating table of the first tree structure, and by changing the content of the relationship-indicator having the node number of each deleted node, the tree structure can be completely changed by merely reassigning the number of the node related to each node when each node under one node is deleted from the existing tree structure . By so doing, the restructuring operation of the tree structure is simplified, reducing the operation time of the operator and the processing time of the computer device.

(0013)

(Embodiment Example)

(Embodiment Example Structure)

Fig. 1 shows a schematic diagram of the data control system having a tree structure, to which the present invention is embodied.

(0014)

The data control system 1 having a tree structure shown in the figure comprises multiple nodes 2 positioned according to the relationships of parent to child and of brother to brother, connection lines 3 connecting between the nodes, and multiple relationship-indicators 4, each created for each node. The node number is assigned to each node providing the uniform meaning to all the numbers. The nodes are grouped by node 2 to form their relationship, for which the relationship-indicators 4 are created; these relationship-indicators are grouped, and the relationship table 10 (See Fig. 3) is created. When new node 2 is added or unnecessary node 2 is deleted, the tree structure 5 can be changed by merely changing the node number corresponding to the node 2 to be added or to be deleted is changed out of the node numbers in the relationship-indicating table 10.

(0015)

In this case, one tree structure 5 is constructed by multiple nodes 2 and multiple connection lines 3 connecting these nodes, and each data element is assigned to each node 2.

(0016)

The relationship-indicator 4 comprises, as shown in Fig. 2, itself - indicating column 6 to write in the node number of itself, parent-indicating column 7 to write in the node number of the node 2 which is a parent to itself, first child-indicating

column 8 to write in the node number of node 2 which is the first child to itself, and member-indicating column 9 to write in the node number of the node 2 which has the most close relationship among the members which share the same parent with itself. Also, the relationship-indicators 4 are grouped and the relationship-indicating table 10 is created, as shown in Fig. 3.

(0017)

(Operation of the Embodiment Example)

The operations of creating the tree structure 5 constituting this data control system and of the relationship-indicating table 10, the insertion operation of new node 2, and the deletion operation of unnecessary node 2 are explained below with reference to the diagrams in Fig. 4 - Fig. 14.

(0018)

(Creating the tree structure and the relationship-indicating table)

When the node number is assigned to each node 2, the node number is assigned to each node 2 constituting the first tree structure a providing the uniform meaning to the nodes, as shown in Fig. 4. At this time, each node number needs not necessarily be consecutive starting from the parent in the topmost position.

(0019)

Subsequently, as shown in Fig. 5, out of the first relationship-indicators 4 in the first relationship-indicating table 10a, the relationship-indicator 4 having the

itself code-indicating column 6, in which its own code number is written, is selected to find the node number of the node 2 which will become its own parent, and this node number is written in the parent-indicating column 7 in its own relationship indicator 4. However, for the topmost node 2, for example, node number "5," "0" is written in as the node number of its own parent.

(0020)

Subsequently, the node number of the node 2 which will become its own first child is found for each node 2, and this node number is written in the first child-indicating column 8 in its own relationship-indicator 4. However, for the nodes 2 in the lowest position, for example, node numbers, "8," "3," "9," "12," "10," and "11," the node number "0" is written in as the node number of its own first child.

(0021)

Then, for each node 2, the node 2 sharing the same parent with itself, in other words, node 2 which will become its own member is found, and the node number of this node 2 is written in the member-indicating column 9 in its own relationship-indicator 4. However, if there are multiple members which share the same parent with itself and itself is in the lowest position in the hierarchical relationship of each member, the node number of the member in the highest position is written in as the node number of the member sharing the same parent with itself. As to the node which is not the member sharing the same parent with itself, for example, node

numbers, "5," "7," and "11," "0" is written as its own member.

(0022)

(New node insertion)

In case when a new tree structure, for example, a tree structure having multiple nodes 2 (e.g., Fig. 6) and multiple connection lines 3 (second tree structure 5b), is inserted into the node 2 having the node number "6" after the first tree structure 5a is constructed by multiple nodes 2 and multiple connection lines 3 in these steps, the node number is assigned to each node 2 constituting the second tree structure 5b in the same steps as those for creating the aforementioned first tree structure 5a and for creating the first relationship-indicating table 10a, as shown in Fig. 7; at the same time, the second relationship-indicating table 10b is created based on the relationship of each node 2, as shown in Fig. 8.

(0023)

Subsequently, as shown in Fig. 9, as a child of the node 2 having the node number "6" constituting the first tree structure 5a, the node 2 positioned in the topmost position of the second tree structure 5b is connected to construct the third tree structure 5c. Then, a portion of the second tree structure 5b constituting the third tree structure 5c is computed to indicate "the new node number of each node 2 constituting the second tree structure 5b = initial node number of itself in the second tree structure 5b + total node number in the first tree structure 5a." The new node

node number of each node constituting the second tree structure 5b is found and, as shown in Fig. 10, a new node number is assigned to each node 2 constituting the second tree structure 5b.

(0024)

As shown in Fig. 11, the initial node number in the initial second relationship-indicating table 10b corresponding to the second tree structure 5b is replaced with a new node number. However, when the node number of the node 2 which becomes its own parent, node number of the node 2 which becomes its own first child, and the node number of the member sharing the same parent with itself are "0," this "0" is kept as is. Subsequently, as shown in Fig. 12, the updated second relationship-indicating 10b is added to the first relationship-indicating table 10a corresponding to the first tree structure 5a, and thus the third relationship-indicating table 10c is created.

(0025)

For the node 2 to which the second tree structure 5b is added, in other words, the node 2 having node number "6," and for each of the nodes 2 having node number "11" and "13," the content of the third relationship-indicating table 10c is changed as follows.

(0026)

First, the node number "11" written in the first child-indicating column 8 in

the relationship-indicator 4 for the node number "6" is changed with node number "13." Then, "0" written in the parent-indicating column 7 in the relationship-indicator 4 for the node number "13" is changed to the node number "6." At the same time, "0" written in the member-indicating column 9 in the relationship-indicator 4 is changed to node number "11."

(0027)

Thereby, when the second tree structure 5b is added to one node 2 constituting the first tree structure 5a, the node number changing operation caused by the insertion of new node 2 can be completed by changing the content of the second relationship-indicating table 10b for the node 2 to which the second tree structure 5b is added, i.e., node 2 having the node number "6," and for the nodes 2 having the node numbers "11" and "13" related to the node number "6."

(0028)

(Deletion operation of unnecessary node 2)

When one of the nodes 2 constituting the first tree structure 5a, i.e. each node 2 positioned under the node 2 having the node number "4," is deleted in said process after having constructed the first tree structure 5a by multiple nodes 2 and multiple connection lines 3, the fourth tree structure 5d is constructed by deleting all the nodes 2 under the node number "4" among the nodes 2 constituting the first tree structure 5a, as shown in Fig. 13.

(0029)

Then, out of the relationship-indicators 4 in the first relationship-indicating table 10a corresponding to the first tree structure 5a, the node numbers of the deleted nodes, i.e., the relationship-indicators 4 corresponding to node numbers, "4," "7," "9," "10," and "12" are all deleted to construct the fourth relationship table 10d shown in Fig. 14, and at the same time, the relationship-indicator 4 having the node number of the deleted node 2 among the node numbers in the fourth relationship-indicating table 10d, e.g., the node number "4" written in the member-indicating column 9 constituting the relationship-indicator 4 of the node 2 having the node number "2," is changed to "0."

(0030)

Accordingly, when each node 2 under the node 2 constituting the first tree structure 5a is deleted, the node number changing operation caused by the deletion of the unnecessary nodes 2 can be performed merely by changing the content of the first relationship-indicating table 10a for the deleted nodes 2 out of the node numbers of the nodes 2 constituting the first tree structure 5a, i.e., the nodes 2 having the node numbers, "4," "7," "9," "10," and "12."

(0031)

(Embodiment Examples)

In this embodiment example, the node numbers having the same meaning are

assigned to all the nodes 2, respectively. At the same time, the relationship of each node is grouped by each node group to create the relationship-indicator 4, and these relationship indicators 4 are put together into the relationship-indicating table 10. Therefore, when new node 2 is added or unnecessary node 2 is deleted, the tree structure 5 can be changed merely by replacing the node number corresponding to the added node 2 or deleted node 2 out of the node numbers in the relationship-indicating table 10. By this, restructuring operation of the tree structure 5 can be simplified, dramatically reducing the operation time of the operator and the processing time of the computer device.

(0032)

In this embodiment example, when new node 2 is inserted into the existing tree structure 5, the tree structure 5 can be changed by merely changing the node number of node 2 and the node number of the node 2 related to this node 2. Thereby, the restructuring operation of the tree structure 5 can be simplified, dramatically reducing the operation time of the operator and the processing time of the computer device.

(0033)

Likewise, when each node 2 positioned under one node 2 from the existing tree structure 5 is deleted, the tree structure 5 can be changed by merely changing the node numbers of the deleted node 2 and of node 2 related to each node 2. By

this, the restructuring operation of the tree structure 5 can be simplified, dramatically reducing the operation time of the operator and the processing time of the computer device.

(0034)

(Advantage)

As explained above, according to Claim 1 of the present invention, when a new node is inserted into or deleted from an existing structure, the tree structure changing operation can be completed merely by changing the node number assigned to each node; thereby simplifying the restructuring operation of the tree structure 5, reducing the operation time of the operator and processing time of computer device.

(0035)

According to Claim 2 of the present invention, when a new node is inserted into or deleted from the existing tree, the tree structure changing operation can be completed by merely changing the numbers of the new node and of the node related to the new node; thereby simplifying the restructuring operation of the tree structure, reducing the operation time of the operator and the processing time of the computer device.

(0036)

According to Claim 3 of the present invention, when each node under one

node is deleted from the existing tree structure, the tree structure change can be completed merely by changing the numbers of the deleted node and of the node related to the deleted node; thereby simplifying the restructuring operation of the tree structure, reducing the operation time of the operator and the processing time of the computer device.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a block diagram of the structure of the data control system having a tree structure into which present invention was embodied.

Fig. 2 shows a schematic diagram of one example of the relationship-indicator constituting the data control system having a tree structure.

Fig. 3 shows a schematic diagram of one example of the relationship-indicating table containing the groups of relationship indicators.

Fig. 4 shows a schematic diagram of the tree structure of Fig. 1 for illustrating the tree structure creating operation of the data control system.

Fig. 5 shows a schematic diagram of the relationship-indicating table of the data control system having a tree structure of Fig. 1 for illustrating the relationship-indicating table creating operation.

Fig. 6 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 7 shows a schematic diagram of the data control system having the tree

structure of Fig. 1 for illustrating the node insertion operation.

Fig. 8 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 9 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 10 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 11 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 12 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node insertion operation.

Fig. 13 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node deletion operation.

Fig. 14 shows a schematic diagram of the data control system having the tree structure of Fig. 1 for illustrating the node deletion operation.

Fig. 15 shows a block diagram of one example of the prior art data control systems having a generally known tree structure.

Fig. 16 shows a block diagram of one example of the data control system when a node number is assigned to each node.

1. Data control system

- 2. Node**
- 3. Connection line**
- 4. Relationship-indicator**
- 5. Tree structure**
 - 5a. First tree structure**
 - 5b. Second tree structure**
 - 5c. Third tree structure**
 - 5d. Fourth tree structure**
- 6. Self-indicating column**
- 7. Parent-indicating column**
- 8. First child-indicating column**
- 9. Member-indicating column**
- 10. Relationship-indicating table**
 - 10a. First relationship-indicating table**
 - 10b. Second relationship-indicating table**
 - 10c. Third relationship-indicating table**
 - 10c. Fourth relationship-indicating table**

Translations
U. S. Patent and Trademark Office
11/01/02
Akiko Smith